

凌云中台服务端入门培训

国内领先的“数字中国”解决方案产品及服务提供商

"DIGITAL CHINA" SOLUTION PRODUCT AND SERVICE PROVIDER

广东飞企互联科技股份有限公司

Guangdong Flying Enterprise Internet Technology Co., Ltd.

密级：内部公开

CONTENTS



工程目录



常用代码



开发规范



部署



日志



<http://api.flyrise.cn:9099/docs/open-docs>

凌云中台开放文档

目录

搜索

基础概念

中台简介

平台概述

平台术语

组织架构

应用套件

物联套件

集成中心

开发技术栈

快速入门

工具组件

开发应用

开放能力

开发规范

常见问题

版本更新

培训讲解

废弃文档

本文档使用 MinDoc 发布

打印

下载

首页

编辑

平台概述

介绍

PGBC业务模组

运营管理

企业服务

人文社区

智能物联

产业生态

第三方

上下游产业链

业务中台

运营中心

应用中心

流程中心

用户中心

API中心

门户中心

数字空间

业务管理

数据中台

数据资产管理

数联中心

数据湖基础设施

技术中台

低代码平台

开发框架

服务治理

基础框架

物联中台

物联中心

边缘计算

协议开发框架

协议调试

云边协同

云原生平台

自动部署

负载均衡

灰度发布

运维中心

服务安全

多租户

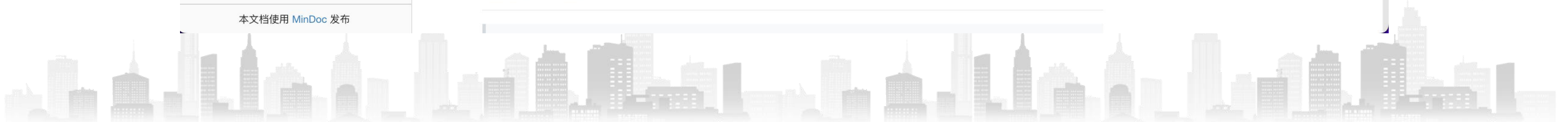
介绍

飞企凭借自身多年智慧园区建设经验进行整理与沉淀，打造出具备智慧园区属性的凌云中台解决方案。在满足园区基本内部管理体系建设的同时，绘制了园区面向产业、生态的产城融合模式运营的蓝图。

为了满足这些业务模式及园区未来的业态发展，凌云中台主要构建了业务中台、数据中台、技术中台、云原生平台等等。

业务中台

- 介绍
- 业务中台
 - 平台运营管理方
 - 平台系统应用开发方
 - 平台系统应用使用方
- 数据中台
 - 数据整合能力
 - 数据开发计算能力
 - 数据服务能力
- 物联中台
 - 北向应用服务
 - 设备协议管理
 - 南向终端管理
- 技术中台
 - 开发运维一体化
 - 开发运营一体化
 - 运行平台
- 云原生平台





- 创
- 使
- 配
- 本

pai c
成功

default

主页

Authorize

Swagger Models

文档管理

凌云中台 - XXX应用

输入文档关键字搜索

凌云中台 - XXX应用

简介

凌云中台 - XXX应用API

作者

朱灿奕

版本

1.0

host

localhost:8080

basePath

/

服务Url

分组名称

default

分组Url

/v2/api-docs

分组location

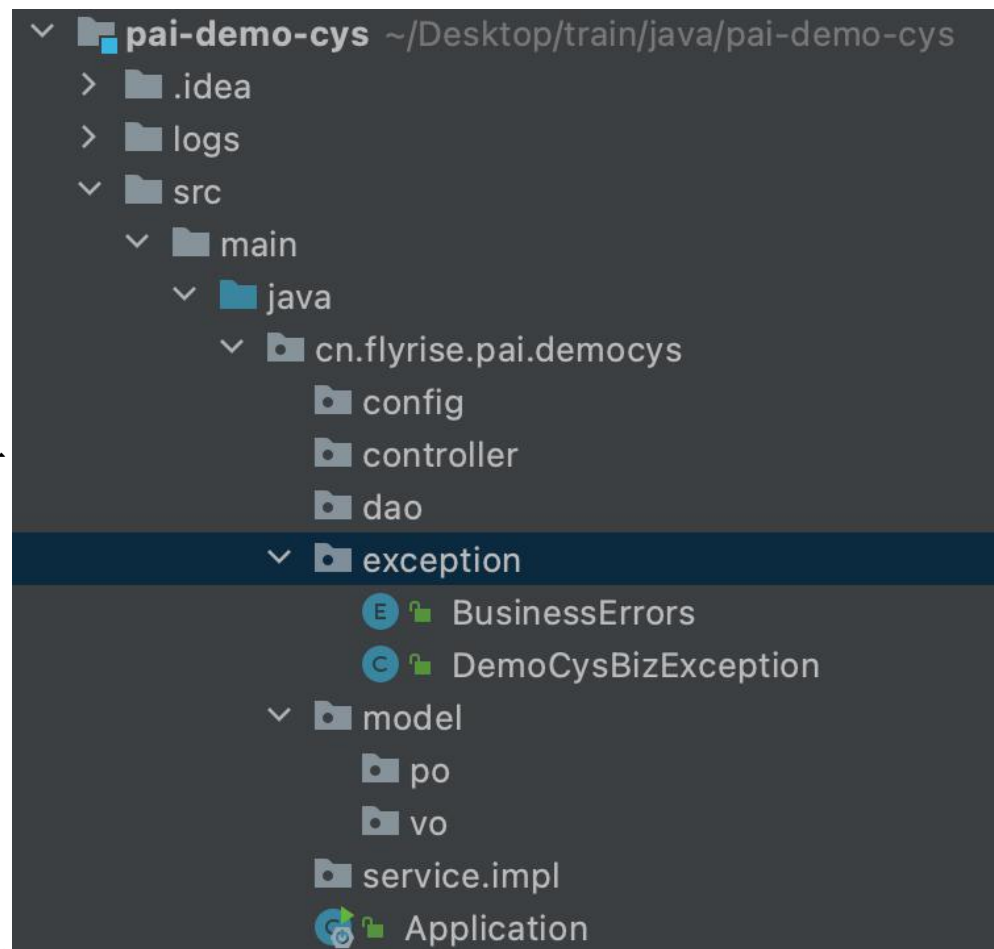
/v2/api-docs

接口统计信息

Apache License 2.0 | Copyright © 2019-Knife4j

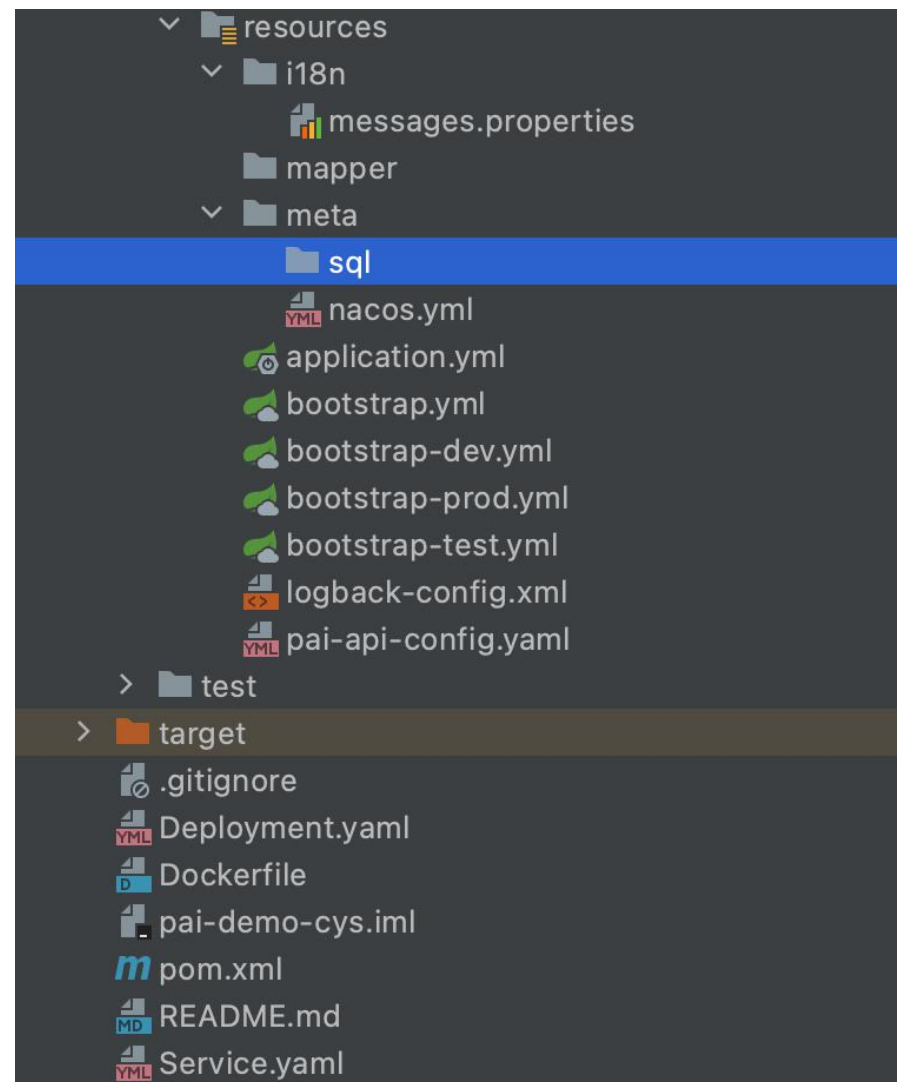


- config:配置类
- controller:控制器，应用程序接收数据/返回数据（或跳转界面）
- dao:数据库访问层，持久层，封装一些对数据库访问的细节
- exception:自定义异常类，异常枚举
- model:领域模型层，po数据库表模型，vo是用于对外暴露的对象
- service:接口类，用来调用领域对象来解决问题
- service.impl:接口实现类，实现定义的接口处理业务逻辑
- Application:项目启动类





- resources:资源目录
- i18n:国际化资源目录
- mapper:mybatis的xml文件，数据库CURD脚本
- meta.sql:数据库脚本，用于边端升级执行的脚本，数据库有变更需要在这个文件添加脚本，脚本规则
V2023022201_add_table.sql
- meta.nacos.yml:边端部署后覆盖nacos配置
- applicaiton.yml:应用特有的配置
- bootstrap.yml:程序引导执行，优于applicaiton.yml
- logback-config:日志配置，已经集成logstash

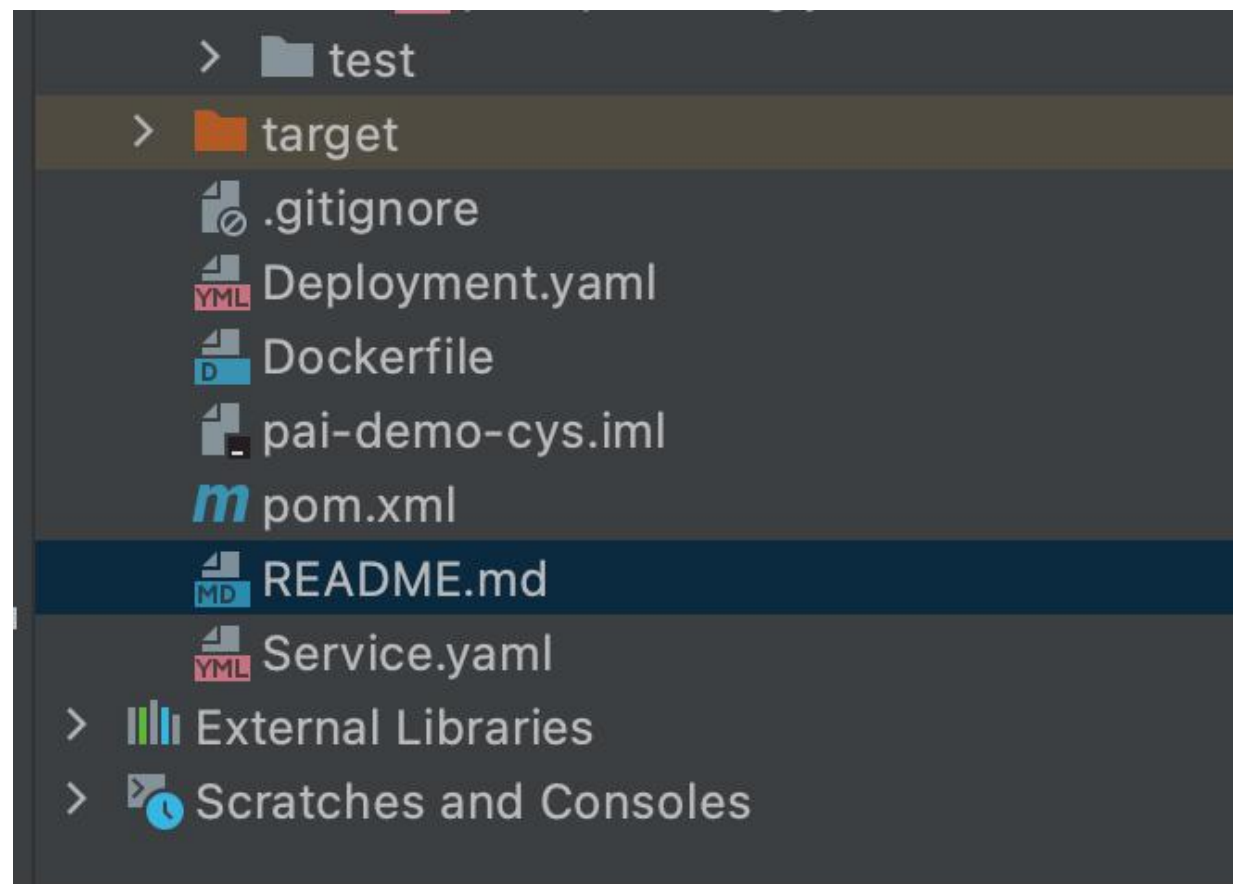




- pom.xml: Maven的依赖管理
- README.md: 对该项目工程的描述

中台部署相关文件，非必要不要调整

- Deployment.yaml: 工作负载的文件
- Dockerfile: 创建docker镜像的文本文件
- Service.yaml: Service的文件



CONTENTS



工程目录



常用代码



开发规范



部署



日志

- `LoginUser loginUser = SecurityUtils.getLoginUser();`//获取当前用户
- `Service.insertBatch(Collection<PO>`
`entityList);Service.updateBatch(Collection<PO> entityList);`//批量新增、修改
- 用户中心API, 组织机构API
- 逻辑删除
- XML热更新
- `@PreAuthorize("@pms.hasSuiteRoles('suiteCode:role:perms'))` //权限控制





临时取消租户条件

- 使在mapper中的方法添加注释@InterceptorIgnore(tenantLine = "true")实现忽略开关

```
@Mapper
@Repository
public interface UserMapper extends BaseMapper<UserPO> {

    @Select("${sql}")
    @ResultType(ArrayList.class)
    @InterceptorIgnore(tenantLine = "true")
    List<UserPO> nativeQuery(@Param("sql") String sql);

    @ResultType(ArrayList.class)
    @InterceptorIgnore(tenantLine = "true")
    List<UserPO> testQuery();

}
```

CONTENTS



工程目录



常用代码



开发规范



部署



日志



IDEA插件安装

- **Alibaba Java Coding Guidelines:** 阿里代码规范检查器
- **SonarLint:** 代码规范与质量检测插件，与阿里代码规范检查器相互补充，帮助写出高质量代码。
- **Save Actions:** IDEA版本如果比较高，会自带，如果没有就需要安装





凌云中采取了GitFlow的模式，并提供多种分支类型。

- 在领到日常功能开发任务时：
 - 简单功能：基于 `develop` 分支直接修改，自测通过后提交代码即可；
 - 复杂功能：基于 `develop` 创建 `feature` 特性开发分支，提交代码后，合并至 `develop` 并删除 `feature` 分支。
- 在领到日常故障修复任务时：
 - 简单故障：基于 `develop` 分支直接修改，自测通过后提交代码即可；
 - 复杂故障：基于 `develop` 创建 `bugfix` 故障修补分支，提交代码后，合并至 `develop` 并删除 `bugfix` 分支。
- 需要发版时，基于 `develop` 创建 `release` 预发测试分支，生成的应用版本部署在UAT测试环境进行测试，期间若需要修改：
 - 简单整改：基于 `release` 上直接处理，自测通过后并提交即可；
 - 复杂整改：基于 `release` 创建 `bugfix` 整改分支，提交代码后，合并至 `release` 并删除 `bugfix` 分支。
- 正式发版时，将 `release` 预发测试分支，合并到 `develop` 和 `master` 分支上，再基于 `master` 打一个标签，如 `v1.0.0.0`，删除 `release` 分支。
- 产品上线后发现故障需要紧急进行热修复时，则基于 `tag` (如 `v1.0.0.0`) 创建 `hotfix` 分支，将修复的代码提交至 `hotfix`；部署该分支上的版本通过验收后，若 `master` 有更新的版本则需要基于 `hotfix` 打出热修版本的 `tag`，如 `v1.0.0.1`，若增加小功能则tag为 `v1.0.1.0`。
- 由于新版本的迭代也同时进行，所以需要在 `hotfix` 上 `develop`，若 `master` 上无更新版本则需要同时合并到 `master`，并基于 `master` 打出热修版本的 `tag`，如 `v1.0.0.1`，若增加小功能则tag为 `v1.0.1.0`，将本次修改的提交合并至 `master`和`develop` 上，完成后删除 `hotfix` 分支。





开发规范 - 分支管理

上线修复
hotfix-*

发布标签
vn.n.n-RELEASE

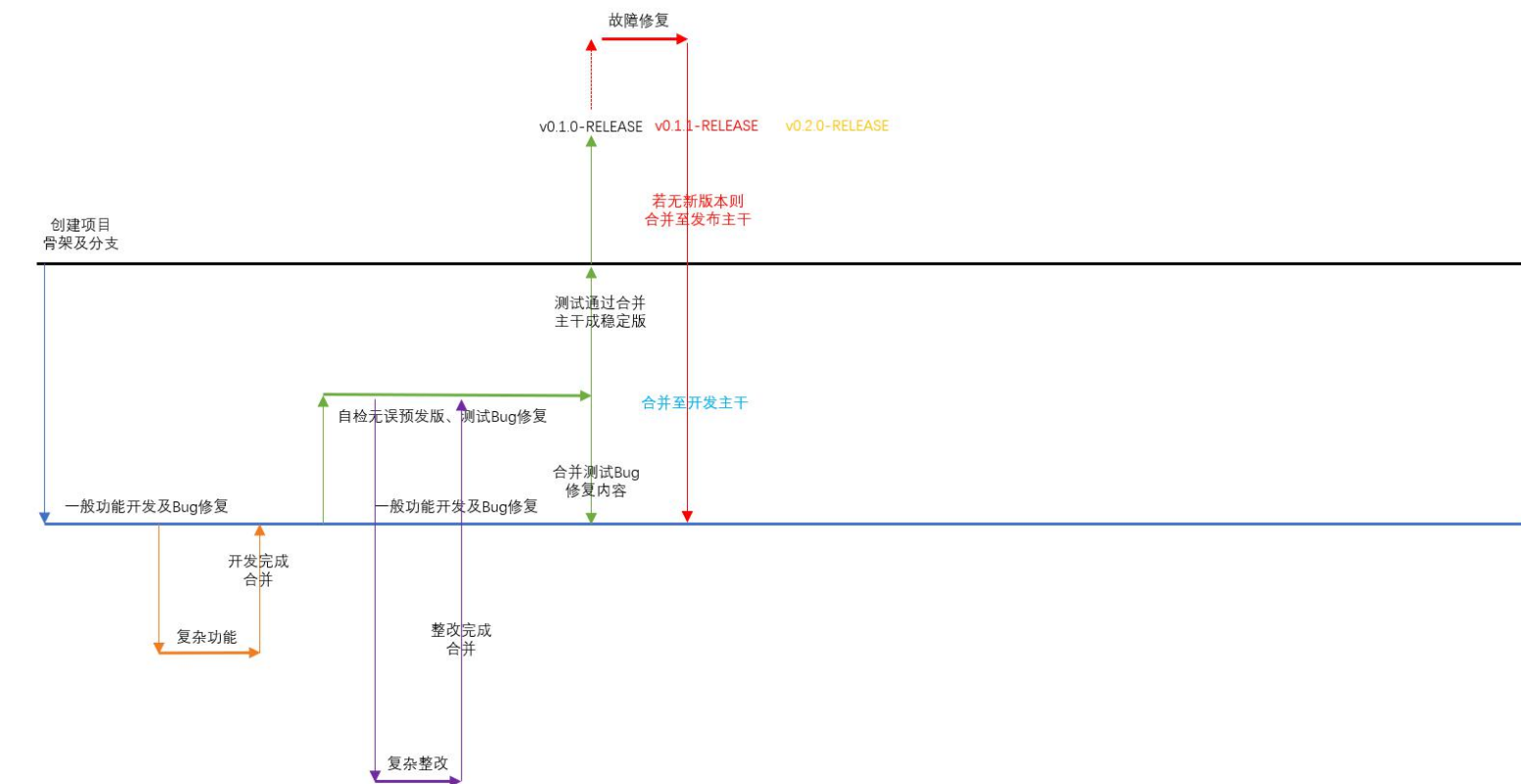
主干分支
master

测试分支
release-*

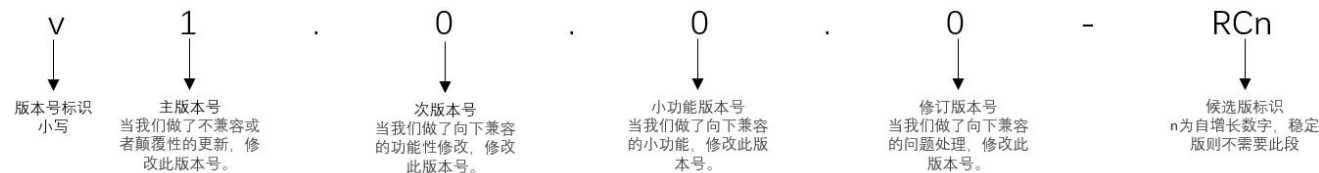
开发分支
develop

特性分支
feature-*

整改分支
bugfix-*



语义化版本:





- 数据库有变更，需要添加脚本至 ->>

/resources/meta/sql

- 脚本规则:

V2023022301__1.0.0_add_table.sql,V2023022302_
_1.1.0_add_data.sql

- 根据场景正确建立索引
- 数据库、表不要设置字符集，使用数据库默认的字符集





- 国际化
- 后端接口返回提示或异常信息等
- 异常处理
- 接口规范
- 接口版本控制及接入版本控制
- 内部调用接口

Java 编码规范



CONTENTS



工程目录



常用代码



开发规范



部署



日志



- 使用pai deploy命令部署到云上
- 通过pai pods 查看运行状态
- 查看Swagger API文档
- 请求API
- 使用pai logs 查看日志
- 使用pai delete命令删除服务



CONTENTS



工程目录



常用代码



开发规范



部署



日志



日志

FE 飞企互联

审计日志:

LINGYUN
运营中心

系统管理

用户管理

角色管理

菜单管理

部门管理

岗位管理

字典管理

参数设置

日志管理

操作日志

平台操作日志

登录日志

首页 / 系统管理 / 日志管理 / 平台操作日志

平台操作日志

日志标题 请输入日志标题 服务ID 请输入服务ID 操作IP 请输入操作IP地址 请求URI 请输入请求URI 类型别名 请输入类型别名 应用名称 请输入应用名称 是否成功 请选择 搜索 重置

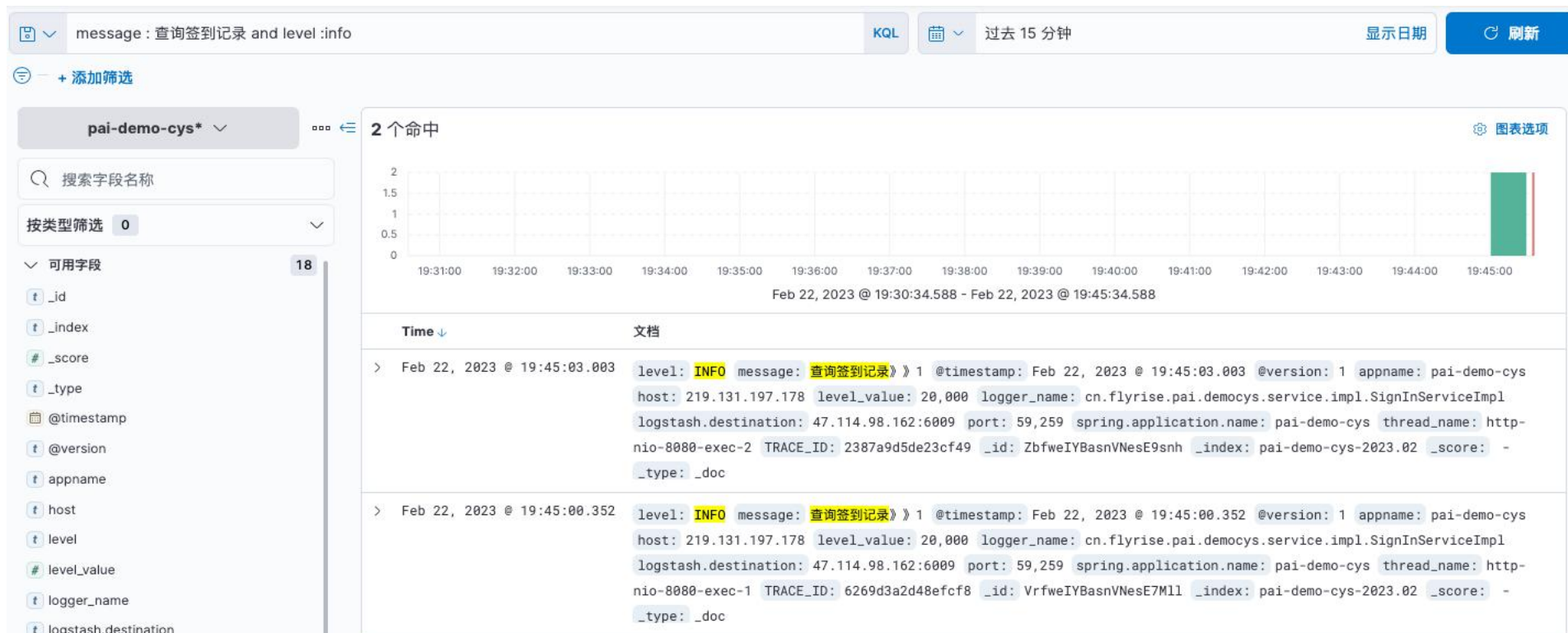
导出

日志类型	日志标题	服务ID	操作IP地址	类型别名	应用名称	成功否	昵称	操作时间	操作
cli	发布表单	tech	120.237.9 9.67	发布表单	pai-tech	是	王晨宇	2023-02-22 19:32:3 4	详细
signAction s	获取单点签到记录	console	0:0:0:0:0: 0:0:1	签到模块	pai-demo -cys	是	测试管理 员	2023-02-22 19:31:4 2	详细
signAction s	获取单点签到记录	console	0:0:0:0:0: 0:0:1	签到模块	pai-demo -cys	是	测试管理 员	2023-02-22 19:31:3 8	详细
cli	发布表单	tech	120.237.9 9.67	发布表单	pai-tech	是	王晨宇	2023-02-22 19:30:3 3	详细
cli	从Kubernetes删除应用	tech	192.168.3. 155	从Kubern etes删除 应用	pai-tech	是	梁焯	2023-02-22 19:30:0 7	详细





- 通过pai logs 查看日志
- 通过[kibana](#) 查看日志





一、套件/工程创建

- 1、创建套件
- 2、创建一个后端工程，并关联套件
- 3、创建一个微应用
- 4、创建一个用户组/权限

二、数据库表创建

- 1、创建一张表
- 2、通过pai code 生成代码

三、编写一个新的接口

- 1、请求参数校验
- 2、swagger注释
- 3、异常处理
- 4、操作日志
- 5、权限控制

四、查看日志

- 1、使用pai logs 查看
- 2、使用kibana查看





**THANK
YOU**





加入培训开发团队:

<https://pai.flyrise.cn/developer/2.0/invite/4abe55769b8a4433ad7313808c72eedc/7c28868e67df79bbd753eb57377c>

加入沙箱企业:

<https://pai.flyrise.cn/console/register/invite/W2JM6Y1V>



开发者账号

